Theoretical Computer Science 411 (2010) 2581–2590

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Connected vertex covers in dense graphs☆

Jean Cardinal *, Eythan Levy

Département d'Informatique, Université Libre de Bruxelles (ULB), CP 212, B-1050 Brussels, Belgium

ARTICLE INFO

Article history:
Received 25 March 2009
Received in revised form 2 March 2010
Accepted 15 March 2010
Communicated by J. Díaz

Keywords:
Approximation algorithm
Vertex cover
Connected vertex cover
Dense graph

ABSTRACT

We consider the variant of the minimum vertex cover problem in which we require that the cover induces a connected subgraph. We give new approximation results for this problem in dense graphs, in which either the minimum or the average degree is linear. In particular, we prove tight parameterized upper bounds on the approximation returned by Savage's algorithm, and extend a vertex cover algorithm from Karpinski and Zelikovsky to the connected case. The new algorithm approximates the minimum connected vertex cover problem within a factor strictly less than 2 on all dense graphs. All these results are shown to be tight. Finally, we introduce the *price of connectivity* for the vertex cover problem, defined as the worst-case ratio between the sizes of a minimum connected vertex cover and a minimum vertex cover. We prove that the price of connectivity is bounded by $2/(1+\varepsilon)$ in graphs with average degree εn, and give a family of near-tight examples.

1. Introduction

The Connected Vertex Cover Problem (CVC) is the variant of Vertex Cover (VC) in which we wish to cover the edges of a given connected graph with a minimum size set of vertices that induce a connected subgraph. The problem was first defined in 1977 by Garey and Johnson [1], who showed it to be NP-hard even when restricted to planar graphs with maximum degree 4. Although CVC has been known for a long time, it has received far less attention than VC until recently.

Regarding approximation algorithms, the first constant ratio is due to Carla Savage [2], who showed that the internal nodes of any depth-first search tree provide a 2-approximation for VC. Such a set of nodes always induces a connected subgraph, and, since the minimum connected vertex cover is always at least as large as the minimum vertex cover, the approximation ratio also applies to CVC. No better constant approximation ratio is known, and recent results [3] have shown that the problem is NP-hard to approximate within less than $10\sqrt{5}-21 \approx 1.36$. Another recent inapproximability result is the APX-hardness of CVC in bipartite graphs [4]. The constant ratio of 2 has recently been improved for several restricted classes of graphs. For instance, Escoffier et al. [4] have shown that CVC is polynomial in chordal graphs, admits a PTAS for planar graphs, and can be approximated within 5/3 for any class of graphs for which VC is polynomial. Approximation results have also been proposed in the field of parallel computing. Fujito and Doi [5] have proposed two parallel 2-approximation algorithms. The first one is an NC algorithm running in time $\mathcal{O}\left(\log^2 n\right)$ using $\mathcal{O}\left(\Delta^2(m+n)/\log n\right)$ processors on an EREW-PRAM, and the second one an RNC algorithm running in $\mathcal{O}(\log n)$ expected time using $\mathcal{O}(m+n)$ processors on a CRCW-PRAM (with n, m and Δ standing for the number of vertices, the number of edges, and the maximum degree, respectively). Several FPT algorithms have also been proposed [6–8], where the parameter is either the size of the optimum or the treewidth.

☆ A preliminary version of this paper appeared in the Proceedings of the 11th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX 2008). This work was partially supported by the *Actions de Recherche Concertées (ARC)* fund of the *Communauté française de Belgique*.

* Corresponding author.
E-mail addresses: jcardin@ulb.ac.be (J. Cardinal), elevy@ulb.ac.be (E. Levy).

0304-3975/$ – see front matter
doi:10.1016/j.tcs.2010.03.021

Density parameters such as the number of edges m and the minimum degree δ have been used in approximation ratios for various optimization problems in graphs (see [9–12] for VC and [16,17,15] for Dominating Set and other problems). These ratios are often expressed as functions of the normalized values of the parameters, namely $m^* = m/\binom{n}{2}$ and $\delta^* = \delta/n$. We call *weakly m^*-dense* and *strongly δ^*-dense* graphs those for which, respectively, m^* and δ^* is a constant. Currently, the best parameterized ratios for VC with parameters m^* and δ^* are $2/(2-\sqrt{1-m^*})$ and $2/(1+\delta^*)$, respectively [9]. Imamura and Iwama [12] later improved the former result, by generalizing it to depend on both m^* and $\Delta^* = \Delta/n$. As for lower bounds, Clementi and Trevisan [13] have proved that VC restricted to strongly dense graphs remained APX-hard. Later, Eremeev [14] showed that it is NP-hard to approximate VC in strongly δ^*-dense graphs within a factor less than $(7+\delta^*)/(6+2\delta^*)$. Finally, Bar-Yehuda et al. [17] prove that if VC cannot be approximated within a factor strictly smaller than 2 on arbitrary graphs, then it cannot be approximated within factors asymptotically smaller than $2/(2-\sqrt{1-m^*})$ and $2/(1+\delta^*)$, respectively, on weakly and strongly dense graphs.

The previous inapproximability results can be shown to hold for the CVC problem as well, using the following trick: to a given input graph G, add a new vertex w, adjacent to all vertices of G. It can be assumed that every minimum solution to the VC problem in this new graph contains w, and hence that it is also a solution for the CVC problem. Furthermore, the normalized minimum degree δ^* and number of edges m^* remain asymptotically the same for both graphs.

Our Results. We present the first parameterized approximation ratios for CVC, with parameters m^* and δ^*. We first analyze Savage's algorithm, and prove approximation ratios of $\min\{2, 1/(1-\sqrt{1-m^*})\}$ and $\min\{2, 1/\delta^*\}$. We then present a variant of Karpinski and Zelikovsy's algorithm which provides better ratios, namely $2/(2-\sqrt{1-m^*})$ and $2/(1+\delta^*)$. These results are the best possible under the assumption that VC cannot be approximated within a factor smaller than 2. However, the algorithm runs in time $\mathcal{O}\left(n^3\right)$ when m^* or δ^* are constant, against the $\mathcal{O}\left(n^2\right)$ complexity of Savage's algorithm.

Finally, we introduce a new graph invariant, the *price of connectivity* for VC, defined as the maximum ratio between the sizes of the optimal solutions of CVC and VC. We prove an upper bound of $2/(1+m^*)$ for the price of connectivity, and present a family of nearly tight graphs.

Notation. We denote by σ and τ the sizes of the optimal solutions of CVC and VC respectively. We denote by m the number of edges in the graph, by δ its minimum degree, and by α the size of its maximum stable set. We use the * notation to denote normalized values of the graph parameters: $\tau^* = \tau/n$, $\sigma^* = \sigma/n$, $m^* = m/\binom{n}{2}$, $\delta^* = \delta/n$ and $\alpha^* = \alpha/n$. We recall that weakly and strongly dense graphs are graphs with bounded values of m^* and δ^*, respectively.

We use the classical notation K_x, I_x and C_x for, respectively, a clique, a stable set and a cycle, on x vertices. We define the *join* $A \times B$ of two graphs A and B as the graph having the edges and vertices of A and B, as well as all possible edges joining both sets of vertices. All graphs considered are assumed to be simple and connected. Throughout the sequel, *OPT* will denote an optimal solution and β the approximation ratio.

2. Savage's algorithm

In 1982, Carla Savage [2] proposed a simple combinatorial algorithm that provides a 2-approximation to VC. It simply returns the internal nodes of an arbitrary depth-first search tree. As this algorithm always returns a connected solution, and $\sigma \geq \tau$, the 2-approximation is also valid for CVC.

Our first lemma provides lower bounds for σ.

Lemma 1. *The following lower bounds hold.*

$$\sigma^* \geq 1 - \sqrt{1-m^*} + \mathcal{O}\left(\frac{1}{n}\right) \tag{1}$$

$$\sigma^* \geq \delta^*. \tag{2}$$

Proof. We consider the first bound. In any graph, since at least $\binom{\alpha}{2}$ edges are missing, we have $m \leq \binom{n}{2} - \binom{\alpha}{2}$; hence $m^* \leq 1 - \alpha^{*2} + \mathcal{O}\left(\frac{1}{n}\right)$. Isolating α^* yields $\alpha^* \leq \sqrt{1-m^*} + \mathcal{O}\left(\frac{1}{n}\right)$. Reverting to the normalized vertex cover $\tau^* = 1 - \alpha^*$ yields $\tau^* \geq 1 - \sqrt{1-m^*} + \mathcal{O}\left(\frac{1}{n}\right)$. As $\sigma \geq \tau$, we obtain the desired result: $\sigma^* \geq 1 - \sqrt{1-m^*} + \mathcal{O}\left(\frac{1}{n}\right)$.

The same kind of reasoning holds for bound (2). In any graph, since a vertex in a maximum stable set has at most $n - \alpha$ neighbors, we have $\delta \leq n - \alpha = \tau$; thus $\tau^* \geq \delta^*$. As $\sigma \geq \tau$, we obtain the desired result: $\sigma^* \geq \delta^*$. □

The upper bounds on the ratio now follow immediately:

Theorem 1. *Savage's algorithm approximates* CVC *within a factor of*

$$\begin{cases} 2 & \text{if } m^* < \frac{3}{4} + o(1) \\ \frac{1}{1-\sqrt{1-m^*}} + o(1) & \text{otherwise.} \end{cases} \quad \text{(weak density)} \tag{3}$$

$$\begin{cases} 2 & \text{if } \delta^* < \frac{1}{2} + o(1) \\ \frac{1}{\delta^*} + o(1) & \text{otherwise.} \end{cases} \quad \text{(strong density)} \tag{4}$$

Proof. Since the ratio of 2 is known from Savage's result, and the value of $n-1$ is the worst possible for any heuristic solution, we trivially have the following bound:

$$\min\left\{2, \frac{n-1}{\sigma}\right\} = \begin{cases} 2 & \text{if } \sigma < \frac{n-1}{2} \\ \frac{n-1}{\sigma} & \text{otherwise.} \end{cases} \tag{5}$$

Normalizing, we get a bound of $\min\left\{2, \frac{1}{\sigma^* + \mathcal{O}\left(\frac{1}{n}\right)}\right\}$. Plugging in inequalities (1) and (2) immediately yields

$$\frac{1}{\sigma^*} \leq \frac{1}{1-\sqrt{1-m^*}+\mathcal{O}\left(\frac{1}{n}\right)} = \frac{1}{1-\sqrt{1-m^*}} + o\left(1\right)$$

$$\frac{1}{\sigma^*} \leq \frac{1}{\delta^*+\mathcal{O}\left(\frac{1}{n}\right)} = \frac{1}{\delta^*} + o\left(1\right).$$

We can now easily compute when the minimum is 2:

$$\frac{1}{1-\sqrt{1-m^*}} + o\left(1\right) > 2 \iff m^* < \frac{3}{4} + o\left(1\right)$$

and

$$\frac{1}{\delta^*} + o\left(1\right) > 2 \iff \delta^* < \frac{1}{2} + o\left(1\right). \quad \square$$

It should be noted that Theorem 1 applies to any 2-approximation algorithm for CVC, as its proof nowhere relies on the specific algorithm being used.

We define the *complete split graph* $\psi_{n,\alpha}$ as the join of a clique $K_{n-\alpha}$ and a stable set I_α. The following lemma expresses the result of Savage's algorithm on complete split graphs.

Lemma 2. *Let H be a worst-case solution returned by Savage's algorithm. Then*

$$|H\left(\psi_{n,\alpha}\right)| = \begin{cases} n-1 & \text{if } \alpha < \frac{n}{2} \\ 2(n-\alpha) & \text{otherwise.} \end{cases}$$

Proof. *Case* 1: $\alpha < \frac{n}{2}$. One possible execution of the algorithm starts from a vertex in the clique, alternatively takes a vertex from the stable set and from the clique, then ends by taking all the remaining vertices in the clique. This execution yields a path of n vertices, and hence a solution of size $n-1$ (by removing the last vertex).

Case 2: $\alpha \geq \frac{n}{2}$. The worst possible execution of the algorithm starts from a vertex in the stable set, then alternatively takes a vertex from the clique and from the stable set. This induces a path of $2(n-\alpha)+1$ vertices, and hence a solution of size $2(n-\alpha)$. $\square$

Theorem 2. *The bounds of Theorem 1 are tight.*

Proof. We show that $\psi_{n,\alpha}$ are tight examples for both bounds (3) and (4).

Optimum. $\tau(\psi_{n,\alpha})$ is trivially $n-\alpha$, and the corresponding optimal solution is the clique $K_{n-\alpha}$. Since this optimal solution is connected, we have $\sigma\left(\psi_{n,\alpha}\right) = \tau\left(\psi_{n,\alpha}\right) = n-\alpha$. Combining this result with the result of Lemma 2 yields

$$\beta\left(\psi_{n,\alpha}\right) = \begin{cases} 2 & \text{if } \alpha > \frac{n}{2} \\ \frac{n-1}{n-\alpha} & \text{otherwise.} \end{cases} = \begin{cases} 2 & \text{if } \sigma^* < \frac{1}{2} + o\left(1\right) \\ \frac{1}{\sigma^*} + \mathcal{O}\left(\frac{1}{n}\right) & \text{otherwise.} \end{cases} \tag{6}$$

Bound 3. Since $\binom{\alpha}{2}$ edges are missing from $\psi_{n,\alpha}$, we have $m(\psi_{n,\alpha}) = \binom{n}{2} - \binom{\alpha}{2}$. Isolating α and normalizing yields $\alpha^*\left(\psi_{n,\alpha}\right) = \sqrt{1-m^*} + \mathcal{O}\left(\frac{1}{n}\right)$. Finally, plugging $\sigma^* = 1-\alpha^* = 1-\sqrt{1-m^*}+\mathcal{O}\left(\frac{1}{n}\right)$ into bound 6 immediately yields bound 3.

Bound 4. It is easy to see that $\delta\left(\psi_{n,\alpha}\right) = n-\alpha = \sigma$. Hence, plugging $\sigma^* = \delta^*$ into bound 6 immediately yields bound 4.

This algorithm runs in time $\mathcal{O}\left(m\right)$, and hence $\mathcal{O}\left(n^2\right)$ for fixed m^*. In the next section, we improve the approximation ratio at the expense of an increase in time complexity.

3. A variant of Karpinski and Zelikovsky's algorithm

Karpinski and Zelikovsky [9] proposed two approximation algorithms that ensure asymptotic approximation ratios of $\frac{2}{1+\delta^*}$ and $\frac{2}{2-\sqrt{1-m^*}}$ respectively. However, they do not always return a connected solution. We propose two variants of their algorithms for CVC, with the same asymptotic approximation factors.

The analysis relies on the following result.

Lemma 3. *Any solution H to* CVC *that consists of*
- a set $H_1 \subseteq OPT$ of size $\epsilon_1 n$,
- a 2-approximation H_2 of CVC *in $G[V - H_1]$ obtained with Savage's algorithm,*
- an additional set H_3 of $\epsilon_2 n$ vertices, with $|H_1| \geq |H_3|$,
approximates CVC *within a factor of*

$$\frac{2}{1+\epsilon_1-\epsilon_2}.$$

Proof. We compute the approximation ratio:

$$\beta = \frac{|H|}{|OPT|} = \frac{|H_1|+|H_2|+|H_3|}{|H_1|+|OPT'|} \quad \text{with } OPT' = OPT - H_1.$$

Note that OPT' is a vertex cover of $G[V - H_1]$, and that H_2 is a 2-approximation of VC in $G[V - H_1]$ (as Savage's algorithm also 2-approximates VC). Hence $|H_2| \leq 2\tau(G[V-H_1]) \leq 2|OPT'|$ and therefore $|OPT'| \geq |H_2|/2$. This yields

$$\beta \leq \frac{|H_1|+|H_2|+|H_3|}{|H_1|+\frac{|H_2|}{2}} = \frac{\epsilon_1 n+|H_2|+\epsilon_2 n}{\epsilon_1 n+\frac{|H_2|}{2}}. \tag{7}$$

Differentiating (7) shows that it grows with $|H_2|$ when $\epsilon_1 \geq \epsilon_2$. Plugging the maximum possible value $|H_2| = n(1-\epsilon_1-\epsilon_2)$ into (7) yields

$$\beta \leq \frac{2}{1+\epsilon_1-\epsilon_2}. \quad \square$$

Algorithm 1 A connected vertex cover algorithm for strongly dense graphs.

procedure P(W) $\qquad \triangleright$ with $W \subseteq V$
  for all vertex $v \in W$ **do**
    $res(v) \leftarrow \{v\} \cup N(v)$
    for all connected components C of $G[V - \{\{v\} \cup N(v)\}]$ **do**
      Find a vertex $c \in C$ that has a neighbor in $N(v)$
      Let $Savage(c)$ be the result of Savage's algorithm in C, starting from c
      $res(v) \leftarrow res(v) \cup Savage(c)$
    end for
  end for
  $v_{\min} \leftarrow \arg\min_{v \in W} |res(v)|$
  return $res(v_{\min})$
end procedure
return $P(V)$

The algorithm of Karpinsky and Zelikovsky makes use of the simple observation that if a vertex does not belong to a vertex cover, then all its neighbors do. Thus for each vertex v, it constructs the vertex cover made of the set $N(v)$ of neighbors of v, and of a 2-approximation on the remaining induced subgraph. Algorithm 1 implements this strategy. To ensure that the returned vertex cover induces a connected graph, we choose to start the execution of Savage's algorithm with a vertex that is connected to $N(v)$.

Theorem 3. *Algorithm 1 approximates* CVC *within a factor of* $\frac{2}{1+\delta^*} + o(1)$.

Proof. It is easy to see that the algorithm returns a connected solution: $\{v \cup N(v)\}$ is connected, and so are the 2-approximations computed in each component C, each of which are connected to $N(v)$ by their starting vertex c. Note that vertex c always exists since the graph is connected.

Furthermore, the returned solution has size at most $|res(v')|$, for some vertex $v' \notin OPT$. Since $v' \notin OPT$, we have $N(v') \subseteq OPT$. Thus Lemma 3 can be applied to $res(v')$, with $|H_1| = |N(v')| \geq \delta^* n$ and $|H_3| = |\{v'\}| = 1$, which immediately yields the desired result. $\square$

The second algorithm is based on the idea of choosing a set of vertices $W \subseteq V$ of size at least ρn, all vertices of which have degree at least ρn for some well-chosen constant ρ. Then either $W \subseteq OPT$, or there exists a vertex w in W such that $N(w) \subseteq OPT$. In either case, a set of size at least ρn is included in OPT, and one can thus try all sets in $\{W \cup \{N(w) : w \in W\}\}$. This original idea of Karpinski and Zelikovsky [9] does not always return a connected solution. In particular, if all vertices of W are in OPT, additional operations are needed, as W does not necessarily induce a connected subgraph. We show that connectivity can be achieved by adding a small set X of carefully chosen vertices (lines 10–14).

Algorithm 2 A connected vertex cover algorithm for weakly dense graphs.

1: $\rho \leftarrow 1 - \sqrt{1 - m^*}$
2: Let W be the set of vertices with degree at least $\rho(n-1)$
3: $C_1 \leftarrow P(W)$ ▷ with $P(\cdot)$ the procedure defined in Algorithm 1
4: $C_2 \leftarrow W$
5: **for all** connected components C of $G[V - W]$ **do**
6: Find a vertex $c \in C$ that has a neighbor in W
7: Let $Savage(c)$ be the result of Savage's algorithm in C, starting from c
8: $C_2 \leftarrow C_2 \cup Savage(c)$
9: **end for**
10: $X \leftarrow \emptyset$
11: **while** $G[W \cup X]$ is not connected **do**
12: Find a vertex v in $V - W$ that is adjacent to the largest number of connected components of $G[W \cup X]$
13: $X \leftarrow X \cup \{v\}$
14: **end while**
15: $C_2 \leftarrow C_2 \cup X$
16: **return** the set of minimum size among C_1 and C_2

The analysis of Algorithm 2 relies heavily on the following lemma, which has been proved in [9].

Lemma 4 ([9]). *Let $\rho = 1 - \sqrt{1 - m^*}$, and let W be the set of vertices with degree at least $\rho(n-1)$. Then W has size at least ρn.*

We now prove an additional property, which upper bounds the number of vertices needed to connect W.

Lemma 5. *There exists a set X of size $\mathcal{O}(\log n)$ such that $G[W \cup X]$ is connected. Such a set is computed in lines 10–14.*

Proof. Let k the total number of components. Note that when $k = \mathcal{O}(\log n)$, the lemma trivially holds; hence we can suppose $k = \omega(\log n)$. We construct X by iteratively choosing a vertex that connects the largest number of remaining connected components.

We note k^j the number of remaining components in $W \cup X$ after j iterations of the loop. We show that, at each step, the algorithm finds a vertex in $V \setminus \{W \cup X\}$ connected to at least $(k^j - 1)\frac{\rho}{1-\rho}$ connected components of $W \cup X$, until $k^j \le \log n$. At step j, we have k^j components, with respective sizes denoted by w_i^j. By definition of W, any vertex in the ith component has degree at least $\rho(n-1)$; hence it has at least $\rho(n-1) - w_i^j + 1$ neighbors in $V - \{W \cup X\}$. Summing over every connected component of $G[W \cup X]$, we get

$$\sum_{i=1}^{k^j} \rho(n-1) - w_i^j + 1 = k^j \rho(n-1) - |W \cup X| + k^j$$
$$= k^j \rho(n-1) - |W| - j + k^j.$$

By the pigeonhole principle, there exists a vertex $v \in V \setminus \{W \cup X\}$ that is connected to at least the following number of components:

$$\frac{k^j \rho(n-1) - |W| - j + k^j}{n - |W \cup X|} = \frac{k^j \rho(n-1) - (|W| + j) + k^j}{n - (|W| + j)}.$$

We may now replace $|W| + j$ by its lower bound of $|W| + j \ge |W| \ge n\rho$ since the fraction $\frac{a-x}{b-x}$ grows with x whenever $a > b$. This condition holds in our case since $a = k^j \rho(n-1) + k^j = \Omega(n \log n) > n = b$, whenever $k^j > \log n$. Thus there exists a vertex $v \in V - W$ that is connected to at least the following number of components:

$$\frac{k^j \rho(n-1) - (|W| + j) + k^j}{n - (|W| + j)} \ge \frac{k^j \rho(n-1) - n\rho + k^j}{n - n\rho}$$
$$= \frac{k^j \rho(n-1) - (n-1)\rho - \rho + k^j}{n - n\rho}$$
$$= \frac{(k^j - 1)\rho(n-1) - \rho + k^j}{n - n\rho}$$

$$
\begin{aligned}
&\geq \frac{(k^j-1)\rho(n-1)-1+k^j}{n-n\rho} \\
&\geq \frac{(k^j-1)\rho(n-1)+(k^j-1)}{n-n\rho} \\
&\geq \frac{(k^j-1)\rho(n-1)+(k^j-1)\rho}{n-n\rho} \\
&= \frac{(k^j-1)\rho n}{n-n\rho} \\
&= \frac{(k^j-1)\rho}{1-\rho}.
\end{aligned}
$$

Note that when $\frac{\rho}{1-\rho} \geq 1$, the algorithm will obtain connectivity with at most two additional vertices, and hence stops earlier than in the opposite case. We now assume $\frac{\rho}{1-\rho} < 1$.

At each step with k^j components in $G[W \cup X]$, a vertex $x \in X$ is thus selected by the algorithm, and connected to at least $\frac{(k^j-1)\rho}{1-\rho}$ connected components. There remain thus $k^{j+1} \leq k^j - \frac{(k^j-1)\rho}{1-\rho} + 1$ components after that step. Hence, the number of components has been divided by

$$
\begin{aligned}
\frac{k^j}{k^{j+1}} &\geq \frac{k}{k - \frac{(k-1)\rho}{1-\rho} + 1} \\
&= \frac{k}{k(1-\frac{\rho}{1-\rho}) + \frac{1}{1-\rho}} \\
&= \frac{1}{(1-\frac{\rho}{1-\rho}) + \frac{1}{k(1-\rho)}} \\
&= \frac{1}{(1-\frac{\rho}{1-\rho}) + o(1)} \quad \text{since } k = \omega(\log n) \\
&= \frac{1}{(1-\frac{\rho}{1-\rho})} - o(1).
\end{aligned}
$$

Since $\frac{\rho}{1-\rho} < 1$, $\frac{1}{(1-\frac{\rho}{1-\rho})} - o(1)$ is at least a constant c strictly greater than 1 for high enough n. Hence, each time a new vertex is added to X, the number of connected components in $G[W \cup X]$ shrinks by a constant factor c. Since the initial number of connected components is at most n, we need at most $\log_c(n) = \mathcal{O}(\log n)$ vertices in X before reaching $k^j \leq \log n$. Then, an additional number of at most $\log n$ vertices is enough to connect the remaining components. Hence

$$|X| \leq \log_c(n) + \log n = \mathcal{O}(\log n). \quad \square$$

Note again that step 6 of the algorithm can always be done; otherwise the graph would not be connected.

Theorem 4. *Algorithm 2 approximates* CVC *within a factor of* $\frac{2}{2-\sqrt{1-m^*}} + o(1)$.

Proof. Two cases can occur. If W contains a vertex $v \notin OPT$, the proof is identical to that of Theorem 4, on plugging $|H_1| = |N(v)| \geq |W|$ and $|H_3| = 1$ into Lemma 3.

On the other hand, if $W \subseteq OPT$, we can again apply Lemma 3 with $|H_1| = |W|$ and, by Lemma 5, $|H_3| = \mathcal{O}(\log n)$. The condition $|H_3| < |H_1|$ required by Lemma 3 holds asymptotically, and we have $\epsilon_2 = \mathcal{O}(\log n)/n \to_{n\to\infty} 0$. Hence the approximation factor is $2/(1+\epsilon_1-\epsilon_2) \to_{n\to\infty} 2/(2-\sqrt{1-m^*})$. $\square$

Theorems 3 and 4 now enable us to state the main corollary:

Corollary 1. CVC *is approximable within a factor strictly less than* 2 *in strongly and weakly dense graphs.*

Note that Algorithms 1 and 2 run in time $\mathcal{O}(nm)$; hence $\mathcal{O}(n^3)$ when m^* is fixed.

Theorem 5. *The bounds of Theorems 3 and 4 are tight.*

Proof. Tightness is witnessed by the following family of graphs: $\nu_{n,\alpha} = K_{n-2\alpha-1} \times (K_1 \times C_{2\alpha})$ (the join of a clique and a "wheel"; see Fig. 1). We first show that $\sigma(\nu_{n,\alpha}) = n - \alpha$ and that both algorithms return $n-1$ on $\nu_{n,\alpha}$. The ratio then follows naturally.

Optimum. One can easily check that taking the clique $K_{n-2\alpha-1}$, the center K_1 of the wheel, and every other vertex of the cycle $C_{2\alpha}$ yields a connected vertex cover of size $n-\alpha$, and that any smaller set would necessarily leave at least one edge uncovered.

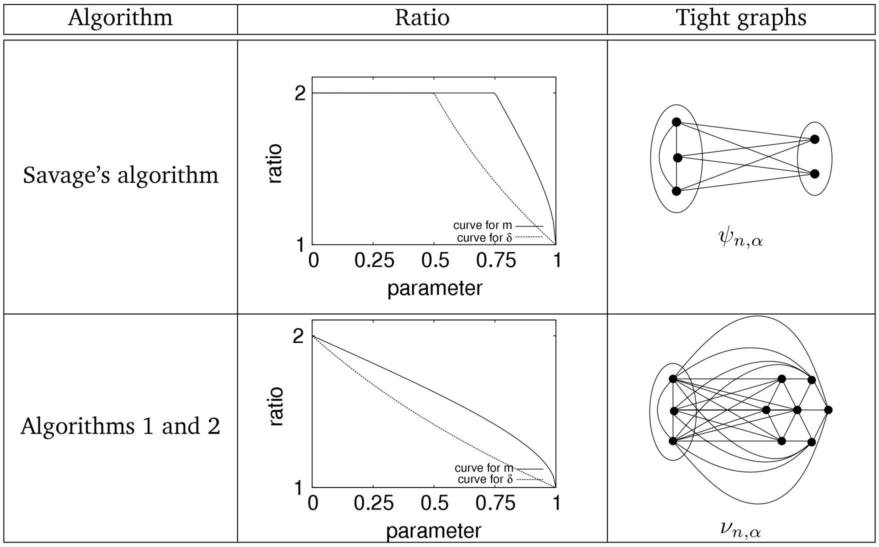


Fig. 1. A comparison of the ratios and tight graphs of Savage's algorithm, Algorithms 1 and 2. The second column compares the asymptotic approximation bounds as functions of parameters m^* and δ^* respectively, while the third column illustrates tight families of graphs for these bounds.

Algorithm 1. If vertex v is in the clique or at the center of the wheel, then $\{\{v\} \cup N(v) = V\}$ and $|res(v)| = n$. If on the other hand v is in the cycle $C_{2\alpha}$, Savage's algorithm is applied in line 6 to only one path $P_{2\alpha-3}$, yielding $|res(v)| = n - 1$.
Algorithm 2. We have $|C_1| = n-1$ for the same reasons as above. Since $\sigma^* \geq 1 - \sqrt{1 - m^*} + o(1)$ (inequality 1), W contains at least the clique and the center of the wheel and hence already induces a connected subgraph. In the worst case, $V - W$ is therefore a path, which implies $|C_2| = n - 1$.
Ratio. Since only the vertices of the cycle $C_{2\alpha}$ have degree less than $n - 1$, we have $\delta(\nu_{n,\alpha}) = n - 2\alpha$. Furthermore, $\nu_{n,\alpha}$ has all possible edges except the $\binom{2\alpha}{2} - 2\alpha$ edges missing from the cycle $C_{2\alpha}$; hence $m(\nu_{n,\alpha}) = \binom{n}{2} - \binom{2\alpha}{2} + 2\alpha$. Solving the given expressions for $\delta(\nu_{n,\alpha})$ and $m(\nu_{n,\alpha})$ for α and inserting the results into our ratio of $(n-1)/(n-\alpha)$ yields the bounds of Theorems 3 and 4. $\square$

The family of graphs described in the above proof also provide tight examples for the original algorithms of Karpinski and Zelikovsky [9], provided they use Savage's algorithm as a subroutine for the 2-approximation phase. In fact, from the result of Bar-Yehuda et al. [17], it is likely that the approximation factors above are the best possible. Fig. 1 summarizes the results of Sections 2 and 3.

A natural question to ask is whether we can use Theorem 1 to boost the approximation ratio of Algorithm 1 and 2. This is not immediately applicable, since we cannot guarantee that the subgraphs $G[V - \{\{v\} \cup N(v)\}]$ (in Algorithm 1) and $G[V - W]$ (in Algorithm 2) remain dense. Nevertheless, Imamura and Iwama [12] managed to apply the idea of Karpinski and Zelikovsky recursively, and obtained a randomized algorithm with a better ratio, depending on both parameters m^* and Δ (maximum degree). We believe this can be done for CVC as well and leave it as future work.

While we have shown that VC and CVC can both be approximated within the same ratio, as a function of m^* or δ^*, this does not settle the question of the price of connectivity, defined as the ratio between the optimal solutions of the two problems.

4. The price of connectivity

In the previous section, we showed that CVC is as well approximable as VC in dense graphs. The question of the maximum ratio between the connected vertex cover and the vertex cover then arises naturally. This notion is relevant in networking applications for which connectivity is a crucial issue. We call this ratio the *price of connectivity*. It is not too difficult to show that the problem of computing the price of connectivity of a given graph is NP-hard (as finding the minimum set of vertices required to connect a given vertex cover is a minimum set cover problem). However, it is not clear that the problem is even in NP.

In this section, we give an upper bound on the price of connectivity for the vertex cover problem in weakly dense graphs, and exhibit a family of graphs whose price of connectivity is close to this upper bound.

4.1. Upper bound

We denote by T an arbitrary optimal vertex cover, by $I = V - T$ the associated maximum stable set, by k the number of connected components in the subgraph induced by T, and by err the difference $\sigma - \tau$. Finally, we denote by S the additional vertices in a smallest connected vertex cover containing T, with size $s = |S|$.

Our first lemma expresses a simple relationship between err, s and k.

Lemma 6. *$err \leq s < k$.*

Proof. The first inequality, err $\leq s$, is straightforward, as any s strictly smaller than err would imply the existence of a connected vertex cover of size smaller than σ. The second inequality, $s < k$, follows from the fact that, since S is a stable set, each one of its vertices necessarily decreases the number of connected components of T by at least 1. □

Our second lemma provides an upper bound on the degrees of the vertices in the maximum stable set I.

Lemma 7. *Every vertex of I is connected to at most $k - s + 1$ different connected components of T.*

Proof. By contradiction, suppose that some vertex v in I is connected to at least $k - s + 2$ connected components of T; then $T \cup \{v\}$ has at most $k - (k - s + 1) = s - 1$ connected components, and hence the smallest subset X of I that contains v and such that $T \cup X$ is connected has size at most $s - 1$, contradicting the minimality of S. □

The last lemma bounds the number of edges by a function of (n, τ, k, s).

Lemma 8. *The following upper bound holds for m:*

$$m \leq \binom{\tau - k + 1}{2} + (n - \tau)(\tau - s + 1). \tag{8}$$

Proof. Let E_1 be the set of edges inside $G[T]$ and E_2 the set of edges between T and I. We bound the size of each set separately.

Clearly, E_1 is maximized when all the connected components in T are cliques. Furthermore, since the total number of edges in those cliques involves a sum of squares, it is maximized with one big clique of size $\tau - k + 1$ and $k - 1$ isolated vertices. Hence $|E_1| \leq \binom{\tau-k+1}{2}$.

We now consider E_2. As each vertex v in I is connected to at most $k - s + 1$ connected components of T (Lemma 7), there are at least $s - 1$ such connected components that v is not connected to, and hence at least $s - 1$ vertices of T that v is not connected to. Hence v cannot have more than $\tau - (s - 1) = \tau - s + 1$ neighbors in T. Multiplying this upper bound of $\tau - s + 1$ by $n - \tau$, the size of I, yields $|E_2| \leq (n - \tau)(\tau - s + 1)$. □

Finally, Theorem 6 follows from first expressing bound 8 as a function of (n, β, τ), then bounding with respect to τ.

Theorem 6. *The ratio between the sizes of a minimum connected vertex cover and a minimum vertex cover in a graph with at least $m^* \binom{n}{2}$ edges is at most $\frac{2}{1+m^*} + o(1)$.*

Proof. Starting from the result of Lemma 8:

$$m \leq \binom{\tau - k + 1}{2} + (n - \tau)(\tau - s + 1),$$

we can see that the bound is a decreasing function of both c and s. We therefore maximize it by taking the smallest possible values for k and s. These values are $s = \text{err}$ and $k = \text{err} + 1$, by Lemma 6. This yields

$$m \leq \binom{\tau - \text{err}}{2} + (n - \tau)(\tau - \text{err} + 1). \tag{9}$$

We define β as the ratio σ/τ. Since err $= \sigma - \tau$, we have err$/\tau = \beta - 1$ and err $= \tau(\beta - 1)$. Plugging this into our last inequality yields

$$m \leq \binom{\tau(2 - \beta)}{2} + (n - \tau)(\tau(2 - \beta) + 1). \tag{10}$$

We now maximize the above expression with respect to τ. Differentiating bound (10) with respect to τ yields a unique maximum at

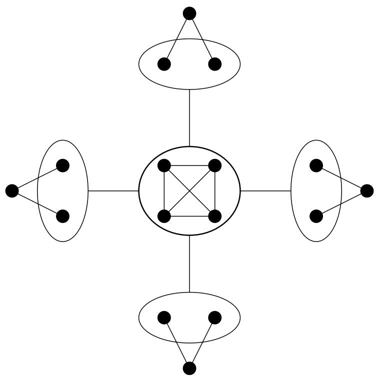
$$\tau_{opt} = \frac{n}{\beta} - \frac{4 - \beta}{2\beta(2 - \beta)} = \frac{n}{\beta} + \mathcal{O}(1) \text{ for each fixed } \beta.$$

Plugging our optimal τ_{opt} into 10 yields

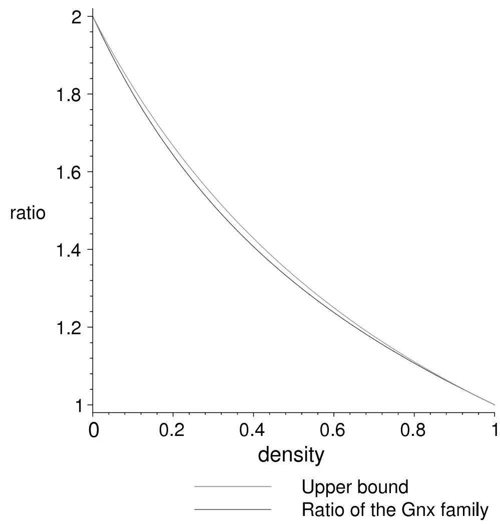
$$m \leq \binom{\tau_{opt}(2 - \beta)}{2} + (n - \tau_{opt})(\tau_{opt}(2 - \beta) + 1) = \frac{n^2(2 - \beta)}{2\beta} + \mathcal{O}(n).$$

Hence

$$m^* \leq \frac{2 - \beta}{\beta} + o(1) \quad \text{and} \quad \beta \leq \frac{2}{1 + m^*} + o(1). \quad \square$$



(a) The graph $G_{16,4}$.



(b) A comparison of the upper bound $2/(1+m^*)$ with the ratio of the class $G_{n,x}$.

Fig. 2. Nearly tight examples.

4.2. Tightness

We now describe a family of graphs whose ratio almost matches the bound of Theorem 6. Let $G_{n,x}$, with $(n-x)$ a multiple of 3, be the graph composed of a clique of size x and $(n-x)/3$ paths P_3, all endpoints of which are totally joined to the clique. Fig. 2(a) illustrates $G_{16,4}$.

The minimum vertex cover consists of the clique of size x and the center of each path, and hence has size $x+(n-x)/3 = (n+2x)/3$. On the other hand, the minimum connected vertex cover consists of the same vertices, augmented with one extra vertex per path, and hence has size $x+2(n-x)/3 = (2n+x)/3$. We therefore have $\beta(G_{n,x}) = \frac{2n+x}{n+2x}$.

We express this bound as a function of the density m^*. The graph $G_{n,x}$ has $\binom{x}{2}$ edges in the clique, $x \cdot 2(n-x)/3$ edges between the paths and the clique, and $2(n-x)/3$ edges in the paths. Hence

$$m(G_{n,x}) = \binom{x}{2} + \frac{x \cdot 2(n-x)}{3} + \frac{2(n-x)}{3} = \frac{2nx}{3} - \frac{x^2}{6} - \frac{7x}{6} + \frac{2n}{3}.$$

Normalizing yields

$$m^*(G_{n,x}) = \frac{m(G_{n,x})}{\binom{n}{2}} = \frac{4x^* - x^{*2}}{3} + \mathcal{O}\left(\frac{1}{n}\right), \quad \text{where } x^* = x/n.$$

Solving the above second-order equation for x^* yields $x^* = 2 \pm \sqrt{4-3m^*} + o(1)$, of which only the solution $x^* = 2 - \sqrt{4-3m^*} + o(1)$ must be kept in order to have x^* in [0, 1].

Plugging this value for x^* into our previous expression for β, it can be checked that

$$\beta(G_{n,x}) = \frac{2n+x}{n+2x} = \frac{4+2m^*+\sqrt{4-3m^*}}{3+4m^*} + o(1).$$

This new bound is very close to the previous one, as shown by Fig. 2(b). In fact, the difference between the old and new ratios does not exceed 1.6% of the latter.

Acknowledgements

The initial conjectures behind several results of this paper were obtained with the help of the GraPHedron software [18] (http://www.graphedron.net), an online tool for obtaining tight inequalities among graph invariants.

References

[1] M.R. Garey, D.S. Johnson, The rectilinear Steiner tree problem is NP complete, SIAM J. Appl. Math. 32 (1977) 826–834.
[2] C.D. Savage, Depth-first search and the vertex cover problem, Inform. Process. Lett. 14 (5) (1982) 233–237.
[3] H. Fernau, D. Manlove, Vertex and edge covers with clustering properties: complexity and algorithms, in: Algorithms and Complexity in Durham 2006 — Proceedings of the Second ACiD Workshop, 18–20 September 2006, Durham, UK, 2006, pp. 69–84.
[4] B. Escoffier, L. Gourvès, J. Monnot, Complexity and approximation results for the connected vertex cover problem, in: Proc. of the 33rd International Workshop on Graph-Theoretic Aspects in Computer Science, WG, 2007, in: LNCS, vol. 4769, Springer-Verlag, 2007, pp. 202–213.

[5] T. Fujito, T. Doi, A 2-approximation NC algorithm for connected vertex cover and tree cover, Inform. Process. Lett. 90 (2) (2004) 59–63.
[6] H. Moser, Exact algorithms for generalizations of vertex cover. Master's thesis, Institut für Informatik, Friedrich-Schiller Universität Jena, 2005.
[7] J. Guo, R. Niedermeier, S. Wernicke, Parameterized complexity of vertex cover variants, Theory Comput. Syst. 41 (3) (2007) 501–520.
[8] D. Mölle, S. Richter, P. Rossmanith, Enumerate and expand: improved algorithms for connected vertex cover and tree cover, Theory Comput. Syst. 43 (2) (2008) 234–253.
[9] M. Karpinski, A. Zelikovsky, Approximating dense cases of covering problems, in: P. Pardalos, D. Du (Eds.), Proc. of the DIMACS Workshop on Network Design: Connectivity and Facilities Location, in: DIMACS series in Disc. Math. and Theor. Comp. Sci., vol. 40, 1997, pp. 169–178.
[10] T. Ibaraki, H. Nagamochi, An approximation of the minimum vertex cover in a graph, Japan J. Indust. Appl. Math. 16 (1999) 369–375.
[11] J. Cardinal, M. Labbé, S. Langerman, E. Levy, H. Mélot, A tight analysis of the maximal matching heuristic, in: Proc. of The Eleventh International Computing and Combinatorics Conference, COCOON, 2005, in: LNCS, vol. 3595, Springer-Verlag, 2005, pp. 701–709.
[12] T. Imamura, K. Iwama, Approximating vertex cover on dense graphs, in: Proc. of the 16th ACM-SIAM Symposium on Discrete Algorithms, SODA, 2005, pp. 582–589.
[13] A.E.F. Clementi, L. Trevisan, Improved non-approximability results for vertex cover with density constraints, in: Proc. of the Second Annual International Conference on Computing and Combinatorics, COCOON, in: LNCS, vol. 1090, Springer-Verlag, 1996, pp. 333–342.
[14] A.V. Eremeev, On some approximation algorithms for dense vertex cover problem, in: Proc. of Symposium on Operations Research., 1999, pp. 48–52.
[15] J. Cardinal, S. Langerman, E. Levy, Improved approximation bounds for edge dominating set in dense graphs, in: Proc. of 4th Workshop on Approximation and Online Algorithms, WAOA, 2006, in: LNCS, vol. 4368, Springer-Verlag, 2006, pp. 108–120.
[16] M. Karpinski, Polynomial time approximation schemes for some dense instances of NP-hard optimization problems, Algorithmica 30 (3) (2001) 386–397.
[17] R. Bar-Yehuda, Z. Kehat, Approximating the dense set-cover problem, J. Comput. Syst. Sci. 69 (4) (2004) 547–561.
[18] H. Mélot, Facet defining inequalities among graph invariants: the system GraPHedron, Discrete Appl. Math. 156 (10) (2008) 1875–1891.